

open



USE



IMPROVE



EVANGELIZE

OpenSolaris 技术介绍

Xinfeng Liu

開
放
的
열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
•••••
πικρ
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை



日程

- Solaris Overview
- Solaris 的特点（系统管理与使用）
- Solaris 上的软件开发
- Solaris 上的性能优化（包括 Dtrace）
- 其它



Solaris 简介

- 市场占有率第一的 Unix
- 支持 Sun Sparc 和 x86 (AMD/Intel)
- 2005 年 6 月 14 日开源
 - CDDL License
 - www.opensolaris.org
 - 1000 万行源代码

Solaris 的主要优势

- 开源
- 历经 20 多年企业应用考验的稳定性
- 优异的性能（尤其是在多 CPU 系统上）
- 完全可抢占的内核
- 先进的系统探测和调试工具
- 安全性
- 可预测的自愈技术
- 支持多种虚拟化技术
- 强大的开发工具的支持
- 文档



Solaris 的特点（系统管理与使用）

- 缺省 Shell (/bin/sh)
- Home 目录 (/export/home)
- 文件系统 (UFS, ZFS, VxFS, QFS...)
- /proc (只包含进程信息)
- 全部都是动态链接的 kernel
- 几个其它目录
 - /usr/ucb, /usr/openwin/bin, /usr/dt/bin
 - **/usr/sfw/bin** (Built-in, Sun supported)
 - /opt/sfw/bin (companion CD, community supported)
 - /usr/xpg4/bin, /usr/xpg6/bin



Solaris 的特点（系统管理与使用）

- 网络配置

- Linux

/etc/inetd.conf

/etc/sysconfig/network-
scripts/ifcfg-xxx

/etc/networks

- Solaris

/etc/inet/inetd.conf

(inetadm, inetconv)

/etc/inet/hosts

/etc/inet/ipnodes

/etc/inet/netmasks

/etc/hostname.<xxx>

/etc/nodename

/etc/dhcp.<xxx>

/etc/defaultrouter

/usr/sbin/sys-unconfig



Solaris 的特点（系统管理与使用）

- 文件系统目录

- Linux

/etc/fstab

/etc/exports

/etc/auto.master

/etc/auto.home

ramdisk

- Solaris

/etc/vfstab

/etc/dfs/dfstab

(/usr/sbin/share)

/etc/auto_master

/etc/auto_home

/tmp, tmpfs, ramdisk



Solaris 的特点（系统管理与使用）

- **SMF (Service Management Facility)**
 - 统一的服务管理，基于 `xml` 的配置文件
 - `svcs -a`
 - `svcadm`
 - `svcs -xv`
 - `svcs -l <service>`
 - `svcs -d <service>`
 - `svcs -D <service>`
 - `svccfg`
 - `svccprop`



Solaris 的特点（系统管理与使用）

- 运行级别 (run level)
 - 0: Power Down (return to OK mode on sparc)
 - s or S or 1: Single User Mode
 - 3: Multiuser Level with NFS resources shared
 - 5: Power-down state (auto turn off)
 - 6: Reboot State
- milestone in SMF
 - 改变缺省的 run level:
`svcs|grep milestone`
`svcadm milestone -d <milestone_name>`

Solaris 的特点（系统管理与使用）

- 常用命令区别
 - <http://bhami.com/rosetta.html>
 - 基本都一样，有少数选项上的差别
 - `/usr/sfw/bin/gtar` VS. `/usr/bin/tar`
 - `tcpdump`, `snoop`
 - `lsof` (from sunfreeware.com)
- 软件包管理
 - `prodreg`,
 - `pkgadd`, `pkgrm`, `pkgchk`
 - `patchadd`, `patchrm`, `showrev -p`
 - `/usr/bin/updatesmanager`, `smpatch`

Solaris 的特点（系统管理与使用）

- **Loopback 设备**

- `lofiadm -a /full-path/xxx.iso`
- `mount -F hsfs /dev/lofi/1 /mnt`

- **刻盘**

- `cdrw -i /full-path/xxx.iso` 从映像文件刻录
- `cdrw -c [-m <tmpdir>]` 复制光盘
- `cdrw -b fast` 抹掉可擦写光盘
- 刻录文件 / 目录

```
mkisofs -o xxx.iso -J -R /full-path-data/
```

```
cdrw -i /full-path/xxx.iso
```



Solaris 的特点（系统管理与使用）

- Kernel 配置
 - /etc/system
 - ndd
 - modinfo, modload, modunload
- 磁盘和卷管理
 - /usr/sbin/format 命令
 - slices
 - X86: 分区 ID 为 0xbf 不再是 0x82 (from Solaris 10 6/2006)
 - 卷管理 :metadb,metainit....



Solaris 的特点（系统管理与使用）

- 设备管理
 - /dev
 - 层次化， /dev/pts/...， /dev/dsk/...
 - /devices
 - /dev/dsk/cAtBdCsD
 - /dev/dsk/cAdBsD
 - /dev/dsk/cAdBpD
 - devfsadm (1M)

Solaris 的特点（系统管理与使用）

- 查看系统信息
 - `prtconf -D` （物理内存，设备树）
 - `prtdiag -v` （系统板及诊断信息）
 - `psrinfo [-v]` (cpu 信息)
 - `iostat -E` , `prtvto` （硬盘几何参数及诊断信息）
 - `isainfo -b` （显示系统 32 位还是 64 位）
 - `isalist` （显示 cpu 支持的指令集）
 - `swap -s` （显示 swap 信息）



Solaris 的特点（系统管理与使用）

- **Fault Manager**

- 自动报告和隔离硬件错误
- `/usr/sbin/fmadm`
- `/usr/sbin/fmstat`
- `/usr/sbin/fmdump`
- <http://www.sun.com/msg/>



Solaris 的特点（系统管理与使用）

- **Fault Manager**

- 例子:

SUNW-MSG-ID: SUN4U-8000-AC, **TYPE:** Fault, **VER:** 1, **SEVERITY:** Major

EVENT-TIME: Thu Feb 26 18:08:26 PST 2004

PLATFORM: SUNW,Sun-Fire-V440, **CSN:** -, **HOSTNAME:** mix

SOURCE: cpumem-diagnosis, **REV:** 0.1

EVENT-ID: 322fe6d5-fe14-6a73-b802-cc6c30b2afcd

DESC: The number of errors associated with this CPU has exceeded acceptable levels. Refer to <http://sun.com/msg/SUN4U-8000-AC> for more information.

AUTO-RESPONSE: An attempt will be made to remove the affected CPU from service.

IMPACT: Performance of this system may be affected.

REC-ACTION: Schedule a repair procedure to replace the affected CPU.

Use **fmdump -v -u <EVENT-ID>** to identify the component to be replaced.

Solaris 的特点（系统管理与使用）

- 安全控制

- RBAC（角色访问控制）

- `/usr/sbin/smc`

- `/etc/user_attr`（用户和角色附加的安全信息）

- `/etc/security/auth_attr`

- `/etc/security/exec_attr`

- `/etc/security/prof_attr`

- `/usr/bin/roles`

- `/usr/bin/auths`

- `/usr/bin/roleadd`



Solaris 的特点（系统管理与使用）

- 安全控制
 - Privileges（进程权限控制）

```
/usr/bin/ppriv -lv
```

```
/usr/bin/ppriv -D
```

```
/usr/bin/ppriv -s
```

```
/usr/bin/usermod -K
```



Solaris 的特点（系统管理与使用）

- 安全控制
 - Trusted Extensions （基于 label 的控制）
 - Solaris Zones （受限的）
 - BART （安全审计）
 - IPSec
 - IP Filter
 - **Secure By Default (/usr/sbin/netservices)**

Solaris 的特点（系统管理与使用）

- 资源控制

- Projects

- /etc/project

- /usr/bin/projects [-1]

- /usr/sbin/projadd

- usermod -K

- /usr/bin/newtask

- id -p

- prstat -J

- prctl （显示或修改进程或项目的资源）



Solaris 的特点（系统管理与使用）

- 资源控制
 - /usr/sbin/rctladm
 - /usr/sbin/rcapadm
 - psrset
 - pooladm
 - Solaris Zone
 - dispadmin
 - priocntl
 - IP QoS (IP classifier)



Solaris 上的开发

- 芯片字节序
 - Sparc: Big Endian
 - X86: Little Endian
- 32 位还是 64 位
 - 哪个性能更好
 - Solaris 64 位 kernel 可同时运行 32 位和 64 位程序
 - 32 位程序的地址空间



Solaris 上的开发

- **IPL32 和 LP64 数据模型**
 - long, pointer, sizeof
 - 尽量显示类型转换以避免符号扩展问题
 - 尽量显示标明常量的类型（如 5L）
 - 尽量使用派生类型从 `<sys/types.h>`, `<inttypes.h>`
如 `time_t`, `uintptr_t`, 以及固定宽度类型如
`int64_t` 等



Solaris 上的开发

- 字节对齐的问题

```
void *p = malloc(10);
```

```
p++;
```

```
int *i = (int *)p;
```

```
*i = 3;
```

在 `sparc` 芯片的机器上

`cc -xarch=v9` 编译



Solaris 上的开发

- 结构 padding 的问题

```
struct bar {  
    int i;  
    long j;  
    int k;  
    char *p;  
};
```

```
cc -xarch=v9
```

```
sizeof(struct bar) ?
```

```
struct bar1 {  
    char *p;  
    long j;  
    int i;  
    int k;  
};
```

```
cc -xarch=v9
```

```
sizeof(struct bar1) ?
```

Solaris 上的开发（移植问题）

- 线程模型
 - Solaris 8, 9, 10
 - Solaris 线程和 Posix 线程 API
- 信号
 - 信号编号, 数量, 缺省 `action`, 以及一些头文件数据结构的区别
- 系统调用
 - `/usr/include/sys/syscall.h`?
- 库函数 (`curses`, `termio`, `ioctl...`)

Solaris 上的开发（移植问题）

- 编译器和 **make** 工具
 - Sun Studio (/opt/SUNWspro/bin)
 - CC, cc
 - dmake
- 编译选项（与 **gcc** 的一些差别）

SUN Studio	GNU GCC
-G	-shared
-xmemalign	-malign-natural
-xO2	-O2
-KPIC	-fPIC
-mt	-pthread



Solaris 上的开发（移植问题）

- 语言扩展
 - Sun CC/cc: `#pragma`
 - gcc: `_attribute_`
- C++ 的移植问题
- LinCAT 移植工具
 - Linux Compatibility Assurance Toolkit
 - 代码扫描工具查找有问题的代码
 - 生成报告
 - 图形界面

Solaris 上的开发

- 链接

- 如何同时链接静态库和动态库

```
cc -o prog main.o -Bstatic -lfoo -Bdynamic -lbar
```

- 链接库的搜索路径

缺省目录: `/usr/ccs/lib, /lib, /usr/lib`

链接选项: `-Lpath1 -llib1... -Lpath2 -llib2...`

`-Rrunpath`

- 环境变量 `LD_LIBRARY_PATH_32` `LD_LIBRARY_PATH_64`

`LD_LIBRARY_PATH` (冒号和分号的差别)

- 搜索顺序



Solaris 上的开发工具 Sun Studio

- 完全免费
 - Sun Studio 11 开始
 - Sun Studio Express
- 高效的 C/C++/Fortran 编译器
- 比 gcc 多得多的优化选项
- 完整的 IDE 开发环境
- 支持 Solaris sparc, Solaris x86 and Linux

Solaris 上的开发工具 Sun Studio

- 性能分析器

★ Performance Analyzer [test.4.er, ...]

File View Timeline Help

Find Text: []

Functions Callers-Callees Source Disassembly Timeline Experiments

User CPU (sec.)	User CPU (sec.)	Source File: /acct/nodel/code/back/src/pubclass/latch.
		Object File: /acct/nodel/code/back/objs/latch.o
		Load Object: <clearlock>
		465. []
		466. return value:0 []
		467. 1 []
		468. */
		469.
0.	0.	470. int CLatch::checklockstatus(latch_lock_t* lockad
		<Function: CLatch::checklockstatus(int*)>
		471. {
		472. #ifdef HP_ENV
		473. return LOCK_TAKEN==(*ALIGNED_ADDR(lockad
		474. #elif SUN_OS
		475.
		476. #ifdef _SPIN_LATCH_
		477. if(SUNSpin_IsLocked(lockaddr))
		478. {
14.200	14.200	479. return LOCK_TAKEN;
		480. }
		481. else
		482. {
		483. return LOCK_FREE;

Summary Event Legend Leak

Data for Selected Object:

Name: line 473 in "latch.cpp"

PC Address: 0:0x00000000

Size: 0

Source File:

Object File:

Load Object:

Mangled Name:

Aliases:

Process Times (sec.) / Counts

	Exclusive	Inclusive
User CPU:	0. (0. %)	0. (0. %)
Wall:	0. (0. %)	0. (0. %)
Total LWP:	0. (0. %)	0. (0. %)
System CPU:	0. (0. %)	0. (0. %)
Wait CPU:	0. (0. %)	0. (0. %)
User Lock:	0. (0. %)	0. (0. %)
Text Page Fault:	0. (0. %)	0. (0. %)
Data Page Fault:	0. (0. %)	0. (0. %)
Other Wait:	0. (0. %)	0. (0. %)

Solaris 上的开发工具 Sun Studio

- 性能分析器

- 先用 `collector` 运行目标程序

```
% /opt/SUNWspro/bin/collect -L unlimited -A copy  
-F on -d /export/home/analyze <cmd>
```

- 然后运行图形化的工具 `analyzer`

```
% cd /export/home/analyze
```

```
% /opt/SUNWspro/bin/analyzer test.3.er
```


Solaris 上的开发工具 Sun Studio

- Open MP 的支持
- Lock_lint 静态源代码分析器对数据和死锁进行追踪
- Thread Analyzer (tha) 检测多线程程序的运行错误
- 命令行和 GUI 调试工具 dbx
- 运行时内存检查工具 RTC

Solaris 上的性能优化（编译器）

- `cc -flags` 查看编译选项
- Sun Studio 一些编译优化选项
 - xarch=xxx (specify the cpu architecture)
 - xO4 (optimization level is 4)
 - fast (a combination of multiple optimizations, be careful for float-pointing codes)
 - lsunperf (using Sun performance library)
 - xipo (cross-file optimization)
 - xlinkopt (optimization during linkage)
 - xautopar -xloopinfo (find and parallel loops)
 - xprefetch_level=3

Solaris 上的性能优化（编译器）

- 基于反馈的优化
 - 编译选项: `-xprofile=collect -x04`
 - 运行代码, 会生成 `profiling` 数据到 `./mycode.xxx`
 - 重新编译: `-xprofile=use:./mycode -x04`
 - 再次运行代码将获得性能提升。
- 源代码优化
 - **Sun Studio** 性能分析器



Solaris 上的性能优化 (cooltools)

- **Sun gcc for sparc**
 - 语法完全兼容 gcc 4.03/4.02
 - 支持 gdb 和 dbx 调试
 - 专为 Sun 的平台优化
 - 更多的优化选项如 `-fast`, `-xipo`, `-xprefetch`, `-xprofile`
 - **CoolTools 完全免费**
<http://cooltools.sunsource.net>



Solaris 上的性能优化 (cooltools)

- **Bit (Binary Improvement Tool)**

- 分析和优化二进制代码 (sparc 平台)

编译: `-xbinopt=prepare`

Instrument: `bit instrument target`

运行 instrument 后的 target: `./target.instr`

分析: `bit analyze target`

优化: `bit optimize -O target`

Solaris 上的性能优化 (cooltools)

- **ATS (Automatic Tuning and Troubleshooting System)**

- 对符合 PEC (Portable Executable Code) 的二进制代码进行优化，无需源码。
- 帮助发现产生错误运行结果的编译选项
- 节约重新编译的时间

前提：编译 'gcc -O3 -xpec' or 'cc -xO3 -Wd,-pec'

例子：

```
ats -i -O4 a.out 自动重编
```

```
ats -i 'script:autotuning' -stopon runs 30 a.out
```

自动实验 30 种不同的编译选项并运行

Solaris 上的性能优化 (cooltools)

- Spot (Simple Performance Optimisation Tool)
 - `spot <cmd>` 或 `spot -P <pid>`
 - 然后将生成 `html` 报告
 - 建议最好用编译选项:
 - `-g` (或 `-g0` for C++) 和 `-xbinopt=prepare` 并且
优化选项至少为 `-x01`



Solaris 上的性能优化 (SAMP)

- **SAMP**

- Solaris + Apache + Mysql + PHP
- 经过 SUN 的专门优化，性能提升 30%-200%
- Solaris sparc and x86
- 完全免费

<http://www.sun.com/software/solaris/amp/>

Solaris 上的性能优化（系统监控）

- prstat
 - 在 solaris 上使用 prstat 而不是 top
 - prstat -a
 - prstat -mL -p <pid>
- Mpstat
- iostat -xn
- vmstat
- psradm
 - 激活或停掉某个 CPU
 - 还可以让某个 CPU 不处理中断请求

Solaris 上的性能优化（系统监控）

- `pstack <pid[/lwps]> or <core>`
 - 打印进程或 coredump 的堆栈信息
- `plimit <pid>`
 - 显示进程的 limits
- `pldd <pid>`
 - 显示进程所使用的库
- `pargs -aec <pid>`
 - 显示进程的参数。比如 Java 虚拟机参数

Solaris 上的性能优化（系统监控）

- `pmap -xs <pid>`
 - 显示进程的内存段
- `pfiles <pid>`
 - 显示进程打开的描述符
- `pwdx <pid>`
 - 显示进程的工作目录
- `psig <pid>`
 - 显示进程的 signal handlers.



Solaris 上的性能优化（系统监控）

- `ptree <pid>`
 - 显示进程的调用树
- `pgrep <match-patterns>`
 - 查找满足条件的进程
- `pkill <match-patterns>`
 - 给满足条件的进程发信号

Solaris 上的性能优化（系统监控）

- `truss <-p pid[/lwps]> or <command>`
 - 跟踪系统调用，库函数的工具
 - `-c`: statistics, `-t`: trace specific system call,
`-u`: trace a library

例子:

```
[paet2k2:/export/home/perfbins]# truss -p 3835
/228:  pollsys(0xFA10FD98, 1, 0xFA10FD20, 0x00000000) = 1
/228:  read(835, " G E T   h t t p : / / w"... , 8191) = 138
/228:  stat("/db0/suncache/s8.013/71/0DD02E1E4D379903", 0x01080B3C) = 0
/228:  open64("/db0/suncache/s8.013/71/0DD02E1E4D379903", O_RDWR) = 766
/228:  read(766, " 0 D D 0 2 E 1 E 4 D 3 7"... , 40) = 40
/228:  read(766, " h t t p : / / w t 2 0 0"... , 40) = 40
.....
```

Solaris 上的性能优化（系统监控）

- cpustat
 - e.g. `#cpustat -c
pic0=L2_dmiss_ld,pic1=Instr_cnt,sys 1 100`
- trapstat
- intrstat
- busstat
- kstat
- mdb
- Lockstat, plockstat

Solaris 上的性能优化 (Dtrace)

- Dtrace
 - 帮助迅速查找系统或应用错误以及性能问题的根源
 - Solaris 系统中内建了数万个 probe
 - 使用动态脚本语言 D 来动态跟踪所希望的东西，可以是各种各样的系统事件和用户事件。超越了以前系统工具的限制。
 - Probe 本身是轻量级的，而且如果没有激活不会对系统产生任何性能影响
 - 可以安全地应用于生产系统

Solaris 上的性能优化 (Dtrace)

D 语言基本格式:

probe description 格式为 *provider:module:function:name*
/ predicate /

```
{  
    action statements  
}
```

例子:

```
#!/usr/sbin/dtrace -s  
syscall::entry  
/execname=="bash"/  
{  
    printf("%s called\n",probefunc);  
}
```


Solaris 上的性能优化 (Dtrace)

- 一些简单的例子

- 为什么我的应用不使用 CPU?

```
# dtrace -n sched:::off-cpu' /execname ==  
"my_app" /{@[ustack()] = count()}'
```

- 我的应用里哪个系统调用执行的最频繁?

```
# dtrace -n syscall:::entry' /pid == 1234 / {  
@syscalls[probefunc] = count(); }'
```

- 跟踪用户的函数调用

```
/usr/demo/dtrace/userfunc.d
```

- 跟踪哪些文件有 I/O 操作

```
/usr/demo/dtrace/iosnoop.d
```

Solaris 上的性能优化 (Dtrace)

- 一些简单的例子

- 跟踪 I/O 时间

`/usr/demo/dtrace/iotime.d`

- 为什么线程 sleep 和 sleep 多长时间？

`/usr/demo/dtrace/whatfor.d`

- 哪些作业在 CPU 的运行队列里？

`/usr/demo/dtrace/whoqueue.d`

- 哪些应用阻塞了多长时间？

`- /usr/demo/dtrace/whofor.d`

Solaris 上的性能优化 (Dtrace)

- 一些简单的例子

- 哪个线程被谁抢占了？

- ```
/usr/demo/dtrace/whopreempt.d
```

- 哪个程序导致了频繁的换页操作？

- ```
dtrace -n pgin' {@[execname] = count()}'
```

- 哪个程序导致了最多的 xcalls？

- ```
dtrace -n xcalls' {@[execname] = count()}'
```

- 更多的例子

# Solaris 上的性能优化 (Dtrace)

- 跟踪我们自己的应用

```
#!/usr/sbin/dtrace -Fs
```

```
pid$target:::entry {}
```

```
pid$target:::return {}
```

还可以这样

```
#!/usr/sbin/dtrace -Fs
```

```
pid$target:mylib:myfunc:entry {}
```

```
pid$target:mylib:myfunc:return {}
```

# Solaris 上的性能优化 (Dtrace)

- 我们的应用哪个函数最花时间

```
#!/usr/sbin/dtrace -s
```

```
pid$target:::entry
```

```
{
```

```
 self->ts[probefunc] = vtimestamp;
```

```
}
```

```
pid$target:::return
```

```
/self->ts[probefunc]/
```

```
{
```

```
 @func_time[probefunc] = sum(vtimestamp-self->ts[probefunc]);
```

```
 self->ts[probefunc] = 0;
```

```
}
```

# Solaris 上的性能优化 (Dtrace)

- 我们的应用哪个函数调用次数最多？

```
#!/usr/sbin/dtrace -s
```

```
pragma D option quiet
```

```
pid$target:::entry
```

```
{
```

```
 @calls[probefunc] = count();
```

```
 /* @calls[ustack()] = count(); */
```

```
}
```

```
dtrace:::END
```

```
{
```

```
 trunc(@calls, 20);
```

```
}
```

# Solaris 上的性能优化 (Dtrace)

- 如何跟踪 java 程序？

```
#!/usr/sbin/dtrace -s
```

```
dvm$1:::method-entry
```

```
{
```

```
 self->interested = 1;
```

```
 self->ts[copyinstr(arg0),copyinstr(arg1),tid] = vtimestamp;
```

```
}
```

```
dvm$1:::method-return
```

```
/ self->interested /
```

```
{
```

```
 @[copyinstr(arg0),copyinstr(arg1),tid] = sum (vtimestamp - self->ts[copyinstr(arg0),copyinstr(arg1),tid]);
```

```
 self->ts[copyinstr(arg0),copyinstr(arg1),tid] = 0;
```

```
}
```

# Solaris 上的性能优化 (Dtrace)

- OpenSolaris 上 Dtrace 最近新增功能

- htons(), htonl(), htonll(), ntohs(), ntohl(), and htohl1() subroutines
- printa() with multiple aggregations
- aggregation sorting options
- umod()/ufunc()/usym(), mod()/func()/sym()
- Chime Visualization Tool for Dtrace
- DTrace support in Zones
- 更多请见:

<http://www.opensolaris.org/os/community/dtrace/ChangeLog/>





# Solaris 上的性能优化 (Dtrace)

- `ufunc()`, `umod()`

```
cat ufunc.d
```

```
#!/usr/sbin/dtrace -qs
```

```
profile-997
```

```
/execname == "bash" && arg1/ {
```

```
 @func[ufunc(arg1)]=count();
```

```
 @module[umod(arg1)]=count();
```

```
}
```

```
END {
```

```
 printf("\nFunctions\n"); printa(@func);
```

```
 printf("Modules\n"); printa(@module);
```

```
}
```



# Solaris 上的性能优化 (Dtrace)

- ucaller

```
-bash-3.00# cat ucaller.d
```

```
#!/usr/sbin/dtrace -s
```

```
pid$target::malloc:entry
```

```
{
```

```
 @[ufunc(ucaller)] = count();
```

```
}
```

```
-bash-3.00# ./ucaller.d -c date
```

```
.....
```

```
ld.so.1`bind_one 1
```

```
libc.so.1`_textdomain_u 1
```

```
ld.so.1`tsort 2
```



# Solaris 上的性能优化 (Dtrace)

- Dtrace 对 Solaris Zone 的支持

```
zonecfg -z myzone
```

```
zonecfg:myzone> set limitpriv=default,dtrace_proc,dtrace_user
```

```
zonecfg:myzone> ^D
```

```
zoneadm -z myzone boot
```

```
zlogin myzone
```

```
myzone# dtrace -l
```

```
myzone# plockstat -Ap `pgrep slapd`
```

还可以从 **global zone** 查看

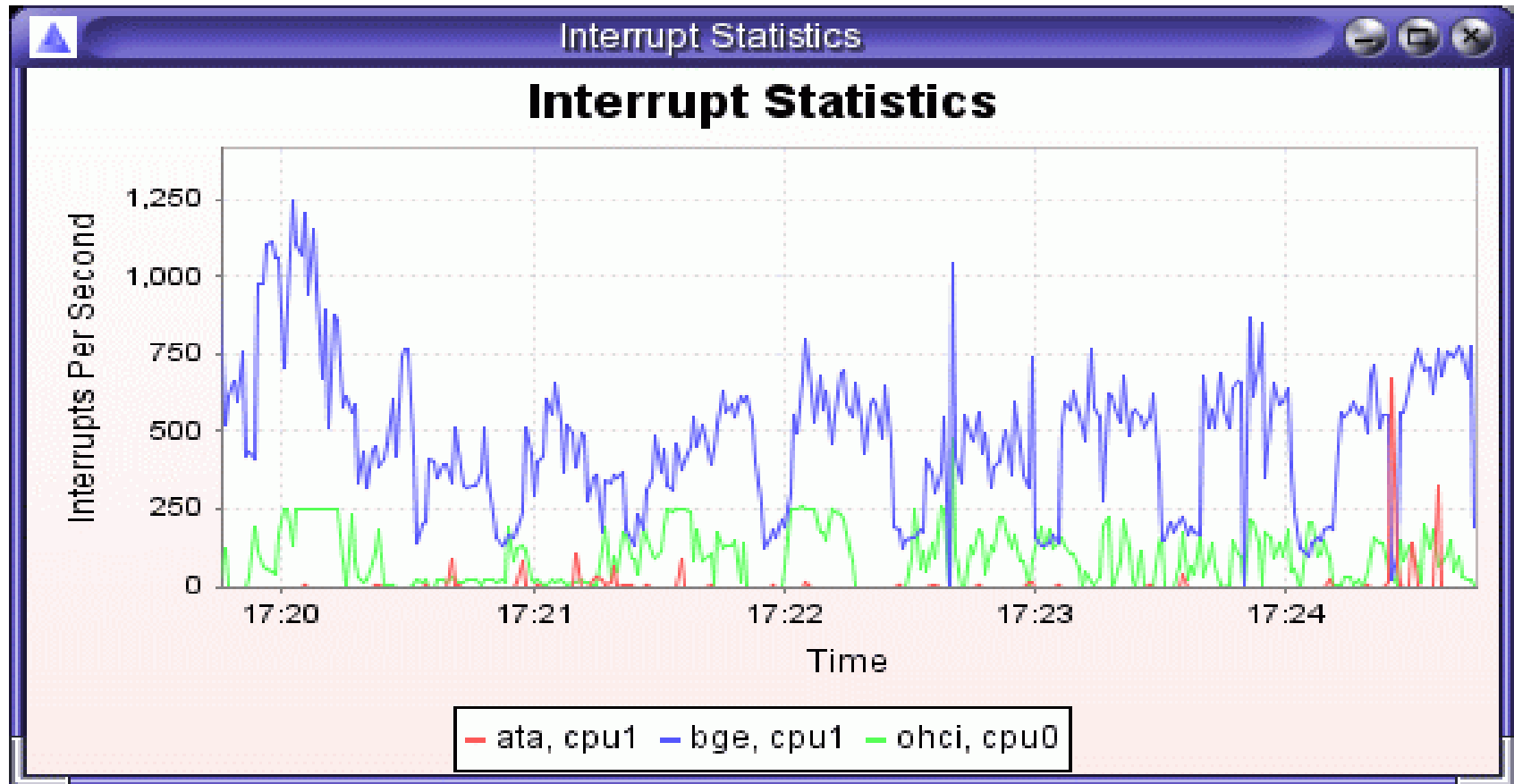
```
dtrace -n io:::start{@[zonename] = count()}
```

# Solaris 上的性能优化 (Dtrace)

- 图形化的 Dtrace

- Dtrace-Chime:

- <http://www.opensolaris.org/os/project/dtrace-chime/>





# Solaris 上查找内存泄漏

- Dbx
  - check-leaks
  - showleaks
- Libumem + mdb
- 查 Java 内存泄漏
  - Java SE5, SE6
  - jmap
  - Heap dump and jhat
  - Hprof
  - jconsole

# Solaris 上查 core dump 原因

- Dbx
  - Dbx <your\_program> <corefile>

dbx 会自动显示出问题的代码

(dbx) threads

(dbx) thread -info <thread\_id>

(dbx) where 显示线程堆栈
- mdb
- pstack <core>



# Solaris 资源链接

- 中文
  - [isv.sun.com.cn](http://isv.sun.com.cn)
  - [developers.sun.com.cn](http://developers.sun.com.cn)
  - [docs.sun.com/app/docs/prod/solaris.10?l=zh](http://docs.sun.com/app/docs/prod/solaris.10?l=zh)
- 英文
  - [www.sun.com/software/solaris](http://www.sun.com/software/solaris)
  - [www.sun.com/software/solaris/howto\\_guides.jsp](http://www.sun.com/software/solaris/howto_guides.jsp)
  - [www.sun.com/software/solaris/learning\\_centers.jsp](http://www.sun.com/software/solaris/learning_centers.jsp)
  - [www.sun.com/bigadmin/products/sol10.html](http://www.sun.com/bigadmin/products/sol10.html)
  - [docs.sun.com](http://docs.sun.com)

# open



USE



IMPROVE



EVANGELIZE

谢谢！

[Xinfeng.Liu@gmail.com](mailto:Xinfeng.Liu@gmail.com)

**ISV Engineering,  
Sun Microsystems Inc.**

“open” artwork and icons by chandan:  
<http://blogs.sun.com/chandan>

開  
放  
的  
열린  
مفتوح  
libre  
मुक्त  
ಮುಕ್ತ  
livre  
libero  
ముక్త  
开放的  
açık  
open  
nyílt  
:::  
πικρ  
オープン  
livre  
ανοικτό  
offen  
otevřený  
öppen  
открытый  
வெளிப்படை